FACSIMILE

WHITE & CASE

White & Case LLP
1155 Avenue of the Americas
New York, New York 10036-2787

Tel + 1 212 819 8200
Fax + 1 212 354 8113
www.whitecase.com

Direct Dial + 212-819-8404 SWeingaertner@whitecase.com

Date:	November 27, 2006	**No. of Pages (including cover):**	52
To:	Commissioner for Patents Mail Stop Appeal Brief - Patents	**Fax Number:** Contact Number:	571-273-8300 571-272-8800
Application No.:	10/066,475	**Filed:**	February 1, 2002
From:	Scott T. Weingaertner	**Reference No.:**	1103179-0009

Attached are the following documents:

(1) Appeal Brief (38 pages); and
(2) Appeal Brief Appendix C attachment (ARM Ltd. v. picoTurbo, Inc. Claim Construction Rulings excerpts) (13 pages).

Respectfully submitted,

Reg. No. 58,992 for

Scott T. Weingaertner, Reg. No. 37,756
(212)-819-8404

CUSTOMER NO. 007470
(212)-819-8200

1103179-0009

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: Edward Colles Nevill
Serial No.	: 10/066,475
Filed	: February 1, 2002
For	: INTEROPERABILITY WITH MULTIPLE INSTRUCTION SETS
Examiner	: Kenneth R. Coulter
Group Art Unit	: 2141

CERTIFICATE OF TRANSMISSION UNDER 37 C.F.R. 1.8
I hereby certify that this paper is being facsimile transmitted to the U.S. Patent and Trademark Office on the date indicated below at the facsimile number 571-273-8300.

Christopher C. Carnaval — Name of Sender
58,992 — Reg. No.
Signature
November 27, 2006 — Date of Signature

Mail Stop Appeal Brief - Patents
Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

FACSIMILE NO: **571-273-8300**
DATE: **November 27, 2006**
PAGES: **38 pages (excluding accompanying papers)**

APPEAL BRIEF

Sir:

Applicant hereby appeals the second rejection of the claims presently pending in the above-referenced reissue application, mailed on February 15, 2006, and sets forth below the bases for this appeal. The Commissioner is hereby authorized to charge any fees associated with the filing of this Appeal Brief to Deposit Account No. 23-1703.

(1) REAL PARTY IN INTEREST

The real party in interest in the above-referenced reissue application is ARM Limited, having a business office at 110 Fulbourn Road, Cambridge CB1 9NJ, United Kingdom.

(2) RELATED APPEALS AND INTERFERENCES

The action ARM Ltd. v. picoTurbo, Inc., C.A. No. 4:00-cv-00957-CW (N.D. Cal.) involved claims for infringement of U.S. Patent No. 6,021,265 ("Original Patent") and six other patents. The matter was resolved pursuant to a consent judgment in which the defendant picoTurbo acknowledged the validity and enforceability of all seven ARM patents. The portions of the Court's claim construction ruling that relate to the Original Patent and the related U.S. Patent No. 5,758,115 are provided in Appendix C. Additional materials related to this action were provided to the Examiner in an Information Disclosure Statement filed on May 2, 2002.

(3) STATUS OF CLAIMS

Claims 1-70 are pending in the reissue application.

Claims 1-14 were originally allowed in Application Serial No. 08/840,557, now U.S. Patent No. 6,021,265, of which the present application is a reissue. Claims 15-64 were added by a Preliminary Amendment filed with the reissue application on February 1, 2002. Claims 65-70 were added by an Amendment originally filed on December 17, 2004, and filed again, on June 3, 2005, in response to a Notice of Non-Compliant Amendment.

This is an appeal from a second, non-final Office Action dated February 15, 2006 ("the 2/15/06 Office Action"), which rejected all pending claims of the above-referenced application.

Claims 1-70 stand rejected under 35 U.S.C. § 102(b) as anticipated by U.S. Patent No. 5,115,500 to Larsen.

Applicant filed a Notice of Appeal and a Pre-Appeal Brief Request for Review on June 15, 2006, with the corresponding fee. Applicant received a Notice of Panel Decision from Pre-Appeal Brief Review, mailed October 25, 2006, indicating that Applicant should proceed to the Board of Patent Appeals and Interferences ("the Board") with its appeal of the rejection of claims 1-70.

(4) STATUS OF AMENDMENTS

No amendments have been submitted subsequent to the mailing of the 2/15/06 Office Action.

(5) SUMMARY OF CLAIMED SUBJECT MATTER

Independent claims 1, 15, 21, 27, 33, 39, 45, 51, 57, 65, 66 and 67, the only independent claims pending in the application, are summarized below in a concise explanation of the subject matter defined by these claims, referring to the original patent specification by column and line number, and to the drawings, as applicable.

A. Independent Claim 1

The data processing apparatus of claim 1 comprises a processor core, a program counter register, logic operable to modify the contents of the program counter, a processor core controller and a memory access controller. The processor core executes successive program instruction words of a predetermined plurality of instruction sets stored in a data memory. The program counter register indicates an address of a next program instruction word in the data memory.

The logic modifies the contents of the program counter register in response to a current program instruction word. The processor core controller, which is responsive to one or more predetermined indicator bits of the program counter register, controls the processor core to execute program instruction words of a current instruction set selected from the predetermined plurality of instruction sets and specified by the state of the one or more indicator bits of the program counter register. The memory access controller accesses program instruction words stored in the data memory, the access controller not being responsive to the one or more indicator bits of the program counter register.

Support for independent claim 1 is provided throughout the specification, including without limitation at column 2, line 1 – column 3, line 2; column 3, lines 10-46; column 3, line 63 – column 4, line 60; column 5, line 1 – column 6, line 4; column 6, lines 23-53; column 7, lines 4-47; and the figures referenced therein.

B. Independent Claim 15

Independent claim 15 is directed to a method of switching between a predetermined plurality of instruction sets used by a data processing apparatus. The claimed method, in response to a first instruction, accesses a sequence of bits, the sequence of bits having an address portion and an instruction set indicator portion; identifies an instruction set selected from the predetermined plurality of instruction sets based on the instruction set indicator portion of the sequence of bits; and sets one or more control flags to indicate that a current instruction set for the data processing apparatus is the instruction set identified based on the instruction set indicator portion of the sequence of bits. The claimed method also retrieves a second instruction from an address derived from the address portion of the sequence of bits. The instruction set

identified by the instruction set indicator portion of the sequence of bits is identifiable without regard to the address derived from the address portion of the sequence of bits.

Support for independent claim 15 is provided throughout the specification, including without limitation at column 2, line 1 – column 3, line 2; column 3, lines 10-46; column 6, lines 23-53; column 7, lines 48-67; and the figures referenced therein.

C. Independent Claim 21

Independent claim 21 is directed to a method of switching between a predetermined plurality of instruction sets used by a data processing apparatus. The claimed method, in response to a first instruction, accesses a sequence of bits, the sequence of bits having an address portion and an instruction set indicator portion, the instruction set indicator portion having at least one bit that is not part of the address portion of the sequence of bits; identifies an instruction set selected from the predetermined plurality of instruction sets based on the instruction set indicator portion of the sequence of bits; and sets one or more control flags to indicate that a current instruction set for the data processing apparatus is the instruction set identified based on the instruction set indicator portion of the sequence of bits. The claimed method also retrieves a second instruction from an address derived from the address portion of the sequence of bits.

Support for independent claim 21 is provided throughout the specification, including without limitation at column 2, line 1 – column 3, line 2; column 3, lines 10-46; column 6, lines 23-53; column 7, lines 48-67; and the figures referenced therein.

D. Independent Claim 27

Independent claim 27 is directed to a data processing apparatus capable of operating using instructions from a predetermined plurality of instruction sets. The data processing apparatus comprises a processor core and a controller. The processor core is responsive to a first

instruction to access a sequence of bits, the sequence of bits having an address portion and an instruction set indicator portion. The processor core derives an address of a second instruction from the address portion of the sequence of bits and uses the instruction set indicator portion of the sequence of bits to set one or more control flags. The controller is responsive to the one or more control flags, the state of the one or more control flags specifying a current instruction set selected from the predetermined plurality of instruction sets, to cause the processor core to execute the second instruction as an instruction from the current instruction set. The one or more control flags are set without regard to the address derived from the address portion of the sequence of bits.

Support for independent claim 27 is provided throughout the specification, including without limitation at column 2, line 1 – column 3, line 2; column 3, lines 10-46; column 4, lines 6-60; column 5, line 1 – column 6, line 4; column 6, lines 23-53; column 7, lines 48-67; and the figures referenced therein.

E. Independent Claim 33

Independent claim 33 is directed to a data processing apparatus capable of operating using instructions from a predetermined plurality of instruction sets. The data processing apparatus comprises a processor core and a controller. The processor core is responsive to a first instruction to access a sequence of bits to derive an address of a second instruction, the sequence of bits having an address portion and an instruction set indicator portion. The instruction set indicator portion of the sequence of bits has at least one bit that is not part of the address portion of the sequence of bits, the address of the second instruction being derived from the address portion of the sequence of bits. The processor core uses the instruction set indicator portion of the sequence of bits to set one or more control flags, the state of the one or more control flags

specifying a current instruction set selected from the predetermined plurality of instruction sets. The controller is responsive to the one or more control flags to cause the processor core to execute the second instruction as an instruction from the current instruction set.

Support for independent claim 33 is provided throughout the specification, including without limitation at column 2, line 1 – column 3, line 2; column 3, lines 10-46; column 4, lines 6-60; column 5, line 1 – column 6, line 4; column 6, lines 23-53; column 7, lines 48-67; and the figures referenced therein.

F. Independent Claim 39

Independent claim 39 is directed to a data processing architecture capable of operating using instructions from a predetermined plurality of instruction sets. The data processing architecture comprises a processor core and a controller. The processor core is responsive to a first instruction to access a sequence of bits, the sequence of bits having an address portion and an instruction set indicator portion. The processor core derives an address of a second instruction from the address portion of the sequence of bits and uses the instruction set indicator portion of the sequence of bits to set one or more control flags. The controller is responsive to the one or more control flags, the state of the one or more control flags specifying a current instruction set selected from the predetermined plurality of instruction sets, to cause the processor core to execute the second instruction as an instruction from the current instruction set. The one or more control flags are set without regard to the address derived from the address portion of the sequence of bits.

Support for independent claim 39 is provided throughout the specification, including without limitation at column 2, line 1 – column 3, line 2; column 3, lines 10-46; column 4, lines

6-60; column 5, line 1 – column 6, line 4; column 6, lines 23-53; column 7, lines 48-67; and the figures referenced therein.

G. Independent Claim 45

Independent claim 45 is directed to a data processing architecture capable of operating using instructions from a predetermined plurality of instruction sets. The data processing architecture comprises a processor core and a controller. The processor core is responsive to a first instruction to access a sequence of bits to derive an address of a second instruction, the sequence of bits having an address portion and an instruction set indicator portion. The instruction set indicator portion of the sequence of bits has at least one bit that is not part of the address portion of the sequence of bits, the address of the second instruction being derived from the address portion of the sequence of bits. The processor core uses the instruction set indicator portion of the sequence of bits to set one or more control flags, the state of the one or more control flags specifying a current instruction set selected from the predetermined plurality of instruction sets. The controller is responsive to the one or more control flags to cause the processor core to execute the second instruction as an instruction from the current instruction set.

Support for independent claim 45 is provided throughout the specification, including without limitation at column 2, line 1 – column 3, line 2; column 3, lines 10-46; column 4, lines 6-60; column 5, line 1 – column 6, line 4; column 6, lines 23-53; column 7, lines 48-67; and the figures referenced therein.

H. Independent Claim 51

Independent claim 51 is directed to a data processing apparatus capable of switching between a predetermined plurality of instruction sets. The data processing apparatus comprises means for accessing a sequence of bits in response to a first instruction, the sequence of bits

having an address portion and an instruction set indicator portion (see, e.g., Original Patent at col. 2, li. 21-26; col. 2, li. 61 - col. 3, li. 2; col. 3, li. 63 – col. 4, li. 25; col. 5, li. 1 – col. 6, li. 4; col. 6, li. 23-53; and the figures referenced therein); means for identifying an instruction set selected from the predetermined plurality of instruction sets based on the instruction set indicator portion of the sequence of bits in response to the first instruction (see, e.g., Original Patent at col. 2, li. 21-26; col. 2, li. 61 – col. 3, li. 2; col. 3, li. 63 – col. 4, li. 25; col. 4, li. 39-53; col. 5, li. 1 – col. 6, li. 4; col. 6, li. 23-53; and the figures referenced therein); means for setting one or more control flags to indicate that a current instruction set for the data processing apparatus is the instruction set identified based on the instruction set indicator portion of the sequence of bits in response to the first instruction (see, e.g., Original Patent at col. 2, li. 21-26; col. 2, li. 61 – col. 3, li. 2; col. 3, li. 63 – col. 4, li. 25; col. 4, li. 39-53; col. 5, li. 1 – col. 6, li. 4; col. 6, li. 23-53; and the figures referenced therein); and means for retrieving a second instruction from an address derived from the address portion of the sequence of bits in response to the first instruction (see, e.g., Original Patent at col. 2, li. 61 – col. 3, li. 2; col. 3, li. 63 – col. 4, li. 25; col. 5, li. 1 – col. 6, li. 4; col. 6, li. 23-53; and the figures referenced therein). The instruction set identified by the instruction set portion of the sequence of bits is identifiable without regard to the address derived from the address portion of the sequence of bits.

Additional support for independent claim 51 is provided throughout the specification, including without limitation at column 2, line 1 – column 3, line 2; column 3, line 10-46; column 3, line 63 – column 4, line 14; column 4, line 39-53; column 5, line 1 – column 6, line 4; column 6, line 23-53; column 7, lines 48-67; and the figures referenced therein.

I. Independent Claim 57

Independent claim 57 is directed to a method of operating a data processing apparatus. The method comprises the steps of receiving a first instruction from a first instruction set selected from a predetermined plurality of instruction sets; translating the first instruction to generate a first set of one or more control signals; accessing a sequence of bits comprising an address portion and an instruction set indicator portion in response to the first set of one or more control signals, the instruction set indicator portion having at least one bit that is not part of the address portion of the sequence of bits; setting one or more control flags based upon the value of the instruction set indicator portion of the sequence of bits to specify that a current instruction set is a second instruction set selected from a predetermined plurality of instruction sets; retrieving a second instruction using an address derived from the address portion of the sequence of bits; and translating the second instruction as an instruction from the current instruction set to generate a second set of one or more control signals.

Support for independent claim 57 is provided throughout the specification, including without limitation at column 2, line 1 – column 3, line 2; column 3, lines 10-46; column 3, line 63 – column 4, line 25; column 4, lines 39-53; column 5, line 1 – column 6, line 4; column 6, lines 23-53; column 7, lines 48-67; and the figures referenced therein.

J. Independent Claim 65

The program counter register of independent claim 65 comprises an ordered set of bits. A subset of the ordered set of bits identifies an address of an instruction. At least one bit of the ordered set of bits identifies an instruction set. The at least one bit is not a member of the subset.

Support for independent claim 65 is provided throughout the specification, including without limitation at column 2, lines 21-26; column 2, line 61 - column 3, line 2; column 3, lines

29-46; column 4, lines 39-53; column 5, lines 1-6; column 5, line 7 – column 6, line 53; column 7, lines 48-67; and the figures referenced therein.

K. Independent Claim 66

Independent claim 66 is directed to a method of selecting an instruction set comprising the steps of receiving a branching instruction written in a first instruction set of a plurality of instruction sets; pursuant to the branching instruction, inserting the address of a next instruction into a register and setting the value of a flag, where the value of the flag is not dependent upon the address of the next instruction; selecting an instruction set based upon the value of the flag; and acquiring the next instruction at the address inserted into the register.

Support for independent claim 66 is provided throughout the specification, including without limitation at column 2, lines 21-26; column 4, lines 6-14; column 6, line 5 - column 7, line 47; and the figures referenced therein.

L. Independent Claim 67

The processing apparatus of independent claim 67 comprises a pointer for identifying an address of a next instruction that is written in a first instruction set of a plurality of instruction sets and a flag for identifying the first instruction set. The pointer and the flag are both written in response to an instruction from a second instruction set of the plurality of instruction sets. The value of the flag is not dependent upon the address of the next instruction.

Support for independent claim 67 is provided throughout the specification, including without limitation at column 2, lines 21-26; column 2, line 61 – column 3, line 2; column 3, lines 29-46; column 4, lines 39-53; column 5, line 1 – column 6, line 53; column 7, lines 48-67; and the figures referenced therein.

(6) GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claims 1-70 are unpatentable under 35 U.S.C. § 102(b) as allegedly anticipated by U.S. Patent No. 5,115,500 to Larsen ("Larsen").

(7) ARGUMENT

CLAIMS 1-70 ARE NOT PROPERLY REJECTED UNDER 35 U.S.C. § 102(b) AS ALLEGEDLY ANTICIPATED BY LARSEN BECAUSE LARSEN IS INCAPABLE OF MEETING AT LEAST ONE LIMITATION OF EACH OF THESE CLAIMS

The rejection of claims 1-70 over Larsen cannot stand because Applicant's invention relies upon the fact that the instruction set is determinable without reference to the address of the instruction (and vice versa) and an instruction of any set may be stored in any location in memory regardless of its instruction type. Whereas, in Larsen, the bits that indicate the instruction type are always part of the instruction address since Larsen relies upon segregating or pre-defining areas of memory to store instructions of a particular instruction set. Thus, far from disclosing all limitations of Applicant's claims, as is required to support an anticipation rejection, Larsen is incapable of operating in the manner of Applicant's invention.

A. SUMMARY

The pending claims are directed to methods and systems for data processing using multiple sets of program instruction words. In rejecting these claims, the Examiner has erred in several respects and has failed to meet his burden of establishing a *prima facie* basis to deny patentability of the claimed invention. The Examiner has mistakenly concluded that the applied reference identically discloses or suggests each and every claim limitation of the present invention, rendering the pending claims anticipated under 35 U.S.C. § 102(b).

Larsen, the reference cited by the Examiner in his rejection of claims 1-70, does not explicitly or even implicitly disclose or suggest each and every claim limitation. In particular, because Larsen relies on the location of an instruction in memory to specify the instruction type, Larsen necessarily must use a portion of the instruction address to specify the current instruction set. Thus, Larsen cannot possibly teach or suggest a system or method wherein a sequence of bits has a portion that specifies the address of the instruction and an independent portion of the sequence that specifies the current instruction set, such that the current instruction set is independent of the location of the instruction in memory. This feature, which distinguishes Applicant's invention from Larsen, is explicitly recited, albeit in different implementations, in each and every claim of Applicant's application.

For example, in claim 1, the memory access controller, which is used to access program instruction words stored in memory, is not responsive to the bits in the program counter that identify the current instruction set; thus, the instruction set is necessarily independent of the instruction address. In claims 15-20 and 51-56 the claims include the limitation that "the instruction set identified by the instruction set indicator portion of bits is *identifiable without regard to the address derived from the address portion of the sequence of bits*." (emphasis added). Claims 21-26, 33-38, 45-50 and 57-64 all require that "the *instruction set indicator portion [has] at least one bit that is not part of the address portion of the sequence of bits*." (emphasis added). Claims 27-32 and 39-44 state that the one or more control flags that specify the instruction set "*are set without regard to the address derived from the address portion of the sequence of bits*." (emphasis added). Claim 65 requires that the at least one bit that identifies the instruction set to be used is "*not a member of the subset*" of bits that specify the instruction address. (emphasis added). Finally, claims 66-70 require setting a flag (e.g., an indicator bit) in

response to a branching instruction "*where the value of the flag is not dependent on the address of the next instruction.*" (emphasis added). Larsen simply cannot disclose or suggest any of these limitations because the system described in Larsen relies on the exactly opposite feature to determine the instruction set; namely, the instruction type is always dependent on the address of the instruction.

Accordingly, Applicant respectfully requests the Board to reverse the rejection of the pending claims.

B. STATEMENT OF APPLICABLE LAW

The Examiner has relied upon 35 U.S.C. § 102(b), reproduced below, to reject pending claims 1-70 in the present application.

> A person shall be entitled to a patent unless – the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of the application for patent in the United States.

"During patent examination the PTO bears the initial burden of presenting a *prima facie* case of unpatentability." In re Glaug, 283 F.3d 1335, 1338 (Fed. Cir. 2002) (citing In re Oetiker, 977 F.2d 1443, 1445 (Fed. Cir. 1992); In re Piasecki, 745 F.2d 1468, 1472 (Fed. Cir. 1984)). See also Ex parte Levy, 17 U.S.P.Q.2d 1461, 1463-64 (Bd. Pat. App. & Interf. 1990) (citation omitted). To establish a *prima facie* case of unpatentability based on anticipation, the PTO must employ a two-step analysis process. See, e.g., Ex parte Babrenburg, Appeal No. 2004-1573, 2005 WL 4773643, at *2-3 (Bd. Pat. App. & Interf. Mar. 31, 2005); Ex parte Monette, Appeal No. 1996-3974, 2001 WL 1255858, at *1 (Bd. Pat. App. & Interf. 2001). First, the USPTO must properly construe the claim terms at issue. See, e.g., Babrenburg, 2005 WL 4773643, at *2-3; Monette, 2001 WL 1255858, at *1 (citing Elmer v. ICC Fabricating, Inc., 67 F.3d 1571, 1574 (Fed. Cir. 1995)). "[W]hen interpreting a claim, words of the claim are generally given their

ordinary and accustomed meaning, unless it appears from the specification or the file history that they were used differently by the inventor." Ex parte Miller, Appeal No. 1997-1238, 1997 WL 1883985, at *4 (Bd. Pat. App. & Interf. 1997) (citing Carroll Touch, Inc. v. Electro Mechanical Sys., Inc., 15 F.3d 1573, 1577 (Fed. Cir. 1993)).

Second, once a claim has been properly construed in the context in which the inventor presented it, it is incumbent upon the USPTO to identify where each and every limitation of the claim is disclosed in a single reference. See Levy, 17 U.S.P.Q.2d at 1462 (citing In re Spada, 15 U.S.P.Q.2d 1655 (Fed. Cir. 1990); In re Bond, 910 F.2d 831, 832 (Fed. Cir. 1990); Lindemann Maschinenfabrik GmbH v. American Hoist and Derrick, 730 F.2d 1452 (Fed. Cir. 1984)). See also Bahrenburg, 2005 WL 4773643, at *3 ("Having construed the claim limitations at issue, we now compare the claims to the prior art to determine if the prior art anticipates those claims." (citations omitted)).

A rejection of a claim as anticipated under 35 U.S.C. § 102(b) requires that each and every claim limitation be identically disclosed in a single prior art reference. See, e.g., Ex parte Nishi, Appeal No. 1997-1381, 1999 WL 33176229, at *2 (Bd. Pat. App. & Interf. Oct. 6, 1999) (quoting RCA Corp. v. Applied Digital Data Systems, Inc., 730 F.2d 1440, 1444 (Fed. Cir. 1984), cert. dismissed, 468 U.S. 1228 (1984)). See also Bahrenburg, 2005 WL 4773643, at *3; Ex parte Sivers, Appeal No. 1997-0730, 1997 WL 1909600, at *2 (Bd. Pat. App. & Interf. 1997) ("It is axiomatic that anticipation of a claim under § 102 can be found only if the prior art reference discloses every element of the claim.") (citing In re King, 801 F.2d 1324, 1326 (Fed. Cir. 1986); Lindemann, 730 F.2d at 1458). Moreover, the USPTO must consider all claim limitations when determining the patentability of an invention over the prior art. See e.g., Bahrenburg, 2005 WL 4773643, at *2 (citing In re Lowry, 32 F.3d 1579, 1582 (Fed. Cir. 1994)).

If the USPTO has not met its burden, a rejection under 35 U.S.C. § 102(b) cannot be sustained and the patent claims must be allowed.

C. THE EXAMINER HAS NOT MET HIS BURDEN OF ESTABLISHING A *PRIMA FACIE* CASE OF UNPATENTABILITY BASED ON ANTICIPATION

The Examiner has failed to demonstrate that the cited Larsen reference identically discloses each and every claim limitation of claims 1-70. Neither the cited passages from Larsen relied on as a basis for the rejection, nor any other passage of that reference, disclose or suggest methods or systems for data processing that use multiple sets of program instruction words wherein the current instruction set is identifiable independent of the address in the instruction. Thus, the Examiner has failed to meet his burden of establishing that each and every claim limitation of claims 1-70 is disclosed by Larsen.

Claims 1-14

The Examiner has rejected claims 1-14 as allegedly anticipated by Larsen but has failed to meet his burden of demonstrating that Larsen teaches or suggests each and every limitation of these claims as required to support a rejection under 35 U.S.C. § 102(b). See, e.g., Nishi, 1999 WL 33176229, at *2. Specifically, Larsen does not, and moreover cannot, disclose a memory access controller that accesses program instruction words stored in data memory while "*not being responsive to the one or more indicator bits of a program counter register.*" In fact, if the system described in Larsen was not responsive to the indicator bits in the program counter, as the Examiner alleges, the system of Larsen would altogether fail to operate because Larsen specifically requires that the instruction set is specified by part of the instruction address (which would be stored in a program counter). Thus, the address resolving logic in Larsen must be responsive to the "indicator" bits of the instruction that specify the instruction set.

Applicant briefly explains the Larsen system with reference to Fig. 2 (reproduced below). In the Larsen system, memory address space is partitioned into pre-defined memory areas to hold instructions from different, respective instruction sets. (See Larsen at Abstract; and col. 5, li. 2-12, 41-51.)

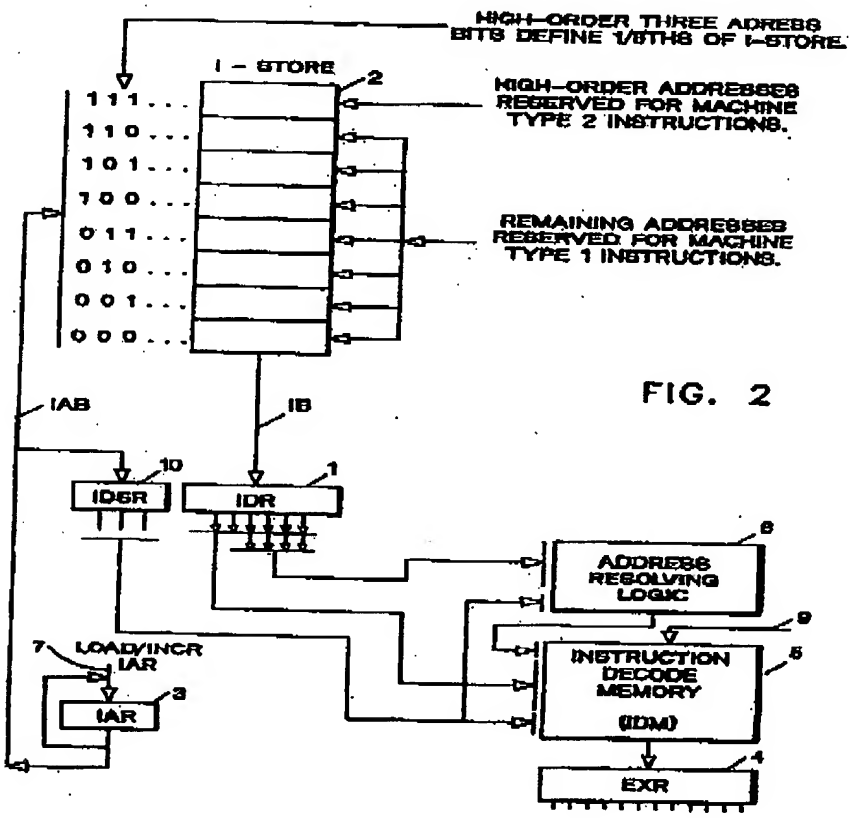

FIG. 2

As Larsen describes and shows in Fig. 2, the register IDSR 10 contains at least the high order bits of the instruction address and the register IDR 1 contains the present instruction to be decoded by the IDM 5. (See Larsen at col. 5, li. 52 – col. 6, li. 16.) As Fig. 2 shows, the instruction decode memory (IDM 5) receives input from both the IDR 1 and the IDSR 10. (See Larsen at col. 6, li. 3-16; and col. 7, li. 12-15.) The decoding of the instruction by the IDM 5 is thus determined in part by the instruction itself (IDR 1) and the location of the instruction in the

instruction store when fetched (IDSR **10**). (See Larsen at Abstract; col. 6, lines 3-16; and col. 7, lines 12-15.) Larsen must decode instructions using both of these inputs because, based on the rigid partitioning of the instruction store (I-store **2**), the high order bits of the instruction address in the IDSR **10** specify the instruction set that is to be used in decoding the instruction. (See Larsen at col. 5, li. 52 – col. 6, li. 16.) Further, the instruction address bits in the IDSR **10** are also used by the address resolution logic **6** as part of the address to access the instruction in question. Thus, in Larsen, the bits in the IDSR **10** are always used both to specify the address of the instruction and the instruction set in which the instruction belongs. (See Larsen at col. 5, li. 34 – col. 6, li. 18.) Therefore, to the extent the Larsen system has a memory access controller (address resolving logic **6** in Larsen), such a controller must be "responsive to" the indicator bits (IDSR **10**) that indicate the current instruction set.

In contrast, independent claim 1 of the present application requires that the memory access controller perform in exactly the opposite manner by "not being responsive to said one or more indicator bits of the program counter register." (See also, e.g., Original Patent at col. 3, li. 42-46; and col. 6, li. 23-53.) Thus, unlike Larsen, in Applicant's invention the location of an instruction in memory and the current instruction set are determined independent of one another.

In light of the above differences between Larsen and the present application, Larsen not only fails to teach or suggest each and every limitation of the independent claims, but it describes a system including a key feature that would fail if implemented in the manner claimed by Applicant. As such, Larsen can not possibly disclose at least subparagraph (v) of independent claim 1, which requires that the memory access controller not be responsive to the one or more indicator bits of the program counter register that specify the instruction set.

This distinction between Larsen and the present application is not trivial. The present application avoids the need for pre-determined partitioning of memory space by a hardware designer as required by Larsen. This provides greater flexibility to a software designer who can partition memory as the designer sees fit or in fact interleave instructions from different instruction sets, according to the needs of the software design. The system of Larsen, on the other hand, does not take into account the actual needs of a particular piece of software in which the amount of memory needed for a particular instruction set will depend on the way in which the software is written. For example, the Larsen system will reserve the highest order one-eighth of the instruction store for "machine type 2" instructions without regard to the amount, if any, of such instructions a software designer wishes to use. If the software designer does not use instructions of this type, or uses fewer than the maximum number of "machine type 2" instructions the partition can store, the unused portions of the partition become a wasted resource that may not be used for storing any other instruction type. Equally problematic, a software designer cannot use more "machine type 2" instructions than can fit in the partition without the use of additional physical memory devices, or without converting some of his code into instructions of a different type, because the excess "machine type 2" instructions cannot be stored anywhere in the remaining partitions reserved for "machine type 1" instructions; otherwise, such instructions would not be properly decoded.

Accordingly, the Examiner has not carried its significant burden of establishing that each and every limitation of the invention set forth in independent claim 1 (and thus claims 2-14 which depend from claim 1) is identically disclosed by Larsen, as required to sustain the rejection. The Examiner has failed to establish a *prima facie* case that Larsen anticipates claims 1-14 because Larsen does not, and cannot, disclose a memory access

controller that accesses program instruction words stored in data memory and is *not responsive to the one or more indicator bits of a program counter register*. Applicant respectfully submits that claims 1-14 are patentable over Larsen.

Claims 15-20 and 51-56

The arguments set forth above with respect to claims 1-14 apply equally to claims 15-20 and 51-56 and are incorporated herein by reference.

Claim 15, from which claims 16-20 depend, and claim 51, from which claims 52-56 depend, each require that "the instruction set identified by the instruction set indicator portion of the sequence of bits is *identifiable without regard to the address derived from the address portion of the sequence of bits*." (emphasis added). As already explained, Larsen cannot possibly teach or suggest this limitation because the address of the instruction, in Larsen, is the determining factor in specifying the instruction set. (See Larsen at col. 6, li. 11-18 ("Decoding of any specific instruction in the IDR 1 thus depends not only on the contents of the IDR 1 but on the region of the I-store 2 from which the instruction was fetched. This portion of the address is provided by the IDSR 10 output. IDSR 10 also has an output path to the address resolution logic 6."). See also, e.g., Larsen at Fig. 2; and col. 5, li. 34 – col. 6, li. 2.) In other words, the address in Larsen always specifies the instruction set.

Accordingly, for the reasons set forth above, the Examiner has failed to establish a *prima facie* case that claims 15 and 51, as well as those claims that depend from them (claims 16-20 and 52-56, respectively), are anticipated by Larsen.

Claims 21-26, 33-38, 45-50 and 57-64

The arguments set forth above with respect to claims 1-14 apply equally to claims 21-26, 33-38, 45-50 and 57-64 and are incorporated herein by reference.

Claim 21, from which claims 22-26 depend, claim 33, from which claims 34-38 depend, claim 45, from which claims 46-50 depend, and claim 57, from which claims 58-64 depend, each require accessing a sequence of bits, the sequence of bits having an address portion and an instruction set indicator portion, "the *instruction set indicator portion having at least one bit that is not part of the address portion of the sequence of bits*." (emphasis added). As already explained, Larsen cannot possibly teach or suggest this limitation because the bits that specify the instruction type, in Larsen, are always part of the address of the instruction. (See Larsen at col. 6, li. 11-18 ("Decoding of any specific instruction in the IDR **1** thus depends not only on the contents of the IDR **1** but on the region of the I-store **2** from which the instruction was fetched. This portion of the address is provided by the IDSR **10** output. IDSR **10** also has an output path to the address resolution logic **6**."). See also, e.g., Larsen at Fig. 2; and col. 5, li. 34 – col. 6, li. 2.)

Accordingly, for the reasons set forth above, the Examiner has failed to establish a *prima facie* case that claims 21, 33, 45 and 57, as well as those claims that depend from them (claims 22-26, 34-38, 46-50 and 58-64, respectively), are anticipated by Larsen.

Claims 27-32 and 39-44

The arguments set forth above with respect to claims 1-14 apply equally to claims 27-32 and 39-44 and are incorporated herein by reference.

Claim 27, from which claims 27-32 depend, and claim 39, from which claims 40-44 depend, each require that the one or more control flags that specify the instruction set "*are set without regard to the address derived from the address portion of the sequence of bits*." (emphasis added). As already explained, Larsen cannot possibly teach or suggest this limitation. In Larsen, the bits that designate the instruction set are always part of the address of the

instruction; thus, no control flag that specifies the instruction set can be set without regard to the instruction address. (See Larsen at col. 6, li. 11-18 ("Decoding of any specific instruction in the IDR 1 thus depends not only on the contents of the IDR 1 but on the region of the I-store 2 from which the instruction was fetched. This portion of the address is provided by the IDSR 10 output. IDSR 10 also has an output path to the address resolution logic 6."). See also, e.g., Larsen at Fig. 2; and col. 5, li. 34 - col. 6, li. 2.)

Accordingly, for the reasons set forth above, the Examiner has failed to establish a *prima facie* case that claims 27 and 39, as well as those claims that depend from them (claims 28-32 and 40-44, respectively), are anticipated by Larsen.

Claim 65

The arguments set forth above with respect to claims 1-14 apply equally to claim 65 and are incorporated herein by reference.

Claim 65 requires an ordered set of bits having (i) a subset of bits that identifies an address of an instruction and (ii) at least one bit that identifies an instruction set, "*wherein the at least one bit is not a member of the subset.*" (emphasis added). As explained above, the specification of Larsen makes clear that Larsen cannot have at least one bit in a program counter that identifies an instruction set but is not part of the bits that are used to identify the instruction address because the instruction set in Larsen is directly dependent on the instruction address. (See Larsen at col. 6, li. 11-18 ("Decoding of any specific instruction in the IDR 1 thus depends not only on the contents of the IDR 1 but on the region of the I-store 2 from which the instruction was fetched. This portion of the address is provided by the IDSR 10 output. IDSR 10 also has an output path to the address resolution logic 6."). See also, e.g., Larsen at Fig. 2; col. 2, li. 42-44; and col. 5, li. 34 – col. 6, li. 2.) To the extent that the system of Larsen has an ordered set of

bits, the bits stored in the IDSR **10** that identify an instruction set are always a subset of the instruction address.

Accordingly, for the reasons set forth above, the Examiner has failed to establish a *prima facie* case that claim 65 is anticipated by Larsen.

Claims 66-70

The arguments set forth above with respect to claims 1-14 apply equally to claims 66-70 and are incorporated herein by reference.

Claim 66, and claim 67, from which claims 68-70 depend, each require identifying the address of a next instruction and setting the value of a flag, wherein "*the value of the flag is not dependent upon the address of the next instruction.*" (emphasis added). As already explained, Larsen cannot possibly teach or suggest this limitation. In Larsen, the bits that designate the instruction set are always part of the address of the instruction; thus, no control flag can be set without regard to the instruction address. (See Larsen at col. 6, li. 11-18 ("Decoding of any specific instruction in the IDR **1** thus depends not only on the contents of the IDR **1** but on the region of the I-store **2** from which the instruction was fetched. This portion of the address is provided by the IDSR **10** output. IDSR **10** also has an output path to the address resolution logic **6**."). See also, e.g., Larsen at Fig. 2; and col. 5, li. 34 – col. 6, li. 2.)

Accordingly, for the reasons set forth above, the Examiner has failed to establish a *prima facie* case that claims 66 and 67, as well as those claims that depend from claim 67 (claims 68-70), are anticipated by Larsen.

(8) CONCLUSION

Applicant respectfully traverses the outstanding rejection over Larsen, because, as explained in detail above, Larsen always uses a portion of the address where an instruction resides to determine the instruction type, whereas in every claim of Applicant's invention, the instruction type is determined independent of the address where the instruction resides.

Applicant respectfully submits that, for the foregoing reasons, Claims 1-70 were improperly rejected as anticipated and are therefore allowable over the cited art. The Examiner has erred. Applicant accordingly requests reversal of the rejections.

Dated: November 27, 2006

Respectfully submitted,

Reg. No. 58,992 for

Scott P. Weingaertner
Reg. No. 37,756
Attorney for Applicant

Customer No. 007470
White & Case LLP
Direct Line: (212) 819-8404

APPENDIX A

APPENDIX OF CLAIMS

1. (Original): Data processing apparatus comprising:

(i) a processor core operable to execute successive program instruction words of a predetermined plurality of instruction sets stored in a data memory;

(ii) a program counter register for indicating an address of a next program instruction word in said data memory;

(iii) logic operable to modify the contents of said program counter register in response to a current program instruction word;

(iv) a processor core controller, responsive to one or more predetermined indicator bits of said program counter register, operable to control said processor core to execute program instruction words of a current instruction set selected from said predetermined plurality of instruction sets and specified by the state of said one or more indicator bits of said program counter register; and

(v) a memory access controller operable to access program instruction words stored in said data memory, said access controller not being responsive to said one or more indicator bits of said program counter register.

2. (Original): Apparatus according to claim 1, comprising:

a first instruction decoder for decoding program instruction words of a first instruction set; and

a second instruction decoder for decoding program instruction words of a second instruction set;

and in which said processor core controller is operable to control either said first instruction decoder or said second instruction decoder to decode a current program instruction word.

3. (Original): Apparatus according to claim 2, in which:

program instruction words of said first instruction set are X-bit program instruction words; and

program instruction words of said second instruction set are Y-bit program instruction words;

Y being diffcrent to X.

4. (Original): Apparatus according to claim 1, in which:

program instruction words of a first instruction set are X-bit program instruction words; and

program instruction words of a second instruction set are Y-bit program instruction words;

Y being different to X.

5. (Original): Apparatus according to claim 3, in which Y is 16 and X is 32.

6. (Original): Apparatus according to claim 4, in which Y is 16 and X is 32.

7. (Original): Apparatus according to claim 1, in which said one or more indicator bits of said program counter register are one or more most significant bits of said program counter register.

8. (Original): Apparatus according to claim 1, in which said one or more indicator bits of said program counter register are one or more least significant bits of said program counter register.

9. (Original): Apparatus according to claim 2, in which said one or more indicator bits of said program counter register are one or more least significant bits of said program counter register.

10. (Original): Apparatus according to claim 3, in which said one or more indicator bits of said program counter register are one or more least significant bits of said program counter register.

11. (Original): Apparatus according to claim 4, in which said one or more indicator bits of said program counter register are one or more least significant bits of said program counter register.

12. (Original): Apparatus according to claim 5, in which said one or more indicator bits of said program counter register are one or more least significant bits of said program counter register.

13. (Original): Apparatus according to claim 6, in which said one or more indicator bits of said program counter register are one or more least significant bits of said program counter register.

14. (Original): Apparatus according to claim 1, comprising a data memory for storing program instruction words to be executed.

15. (Previously presented): A method of switching between a predetermined plurality of instruction sets used by a data processing apparatus, the method comprising:

in response to a first instruction:

(i) accessing a sequence of bits, the sequence of bits having an address portion and an instruction set indicator portion;

(ii) identifying an instruction set selected from the predetermined plurality of instruction sets based on the instruction set indicator portion of the sequence of bits;

(iii) setting one or more control flags to indicate that a current instruction set for the data processing apparatus is the instruction set identified based on the instruction set indicator portion of the sequence of bits; and

retrieving a second instruction from an address derived from the address portion of the sequence of bits,

wherein the instruction set identified by the instruction set indicator portion of the sequence of bits is identifiable without regard to the address derived from the address portion of the sequence of bits.

16. (Previously presented): The method of claim 15, further comprising executing the second instruction as an instruction of the current instruction set.

17. (Previously presented): The method of claim 15 in which the predetermined plurality of instruction sets comprises a first instruction set and a second instruction set, and wherein

instructions of the first instruction set are X-bit instructions and instructions of the second instruction set are Y-bit instructions, where Y is different from X.

18. (Previously presented): The method of claim 17 wherein X is 32 and Y is 16.

19. (Previously presented): The method of claim 15 wherein the instruction set indicator portion of the sequence of bits comprises one or more least significant bits of the sequence of bits.

20. (Previously presented): The method of claim 15 wherein the instruction set indicator portion of the sequence of bits comprises one or more most significant bits of the sequence of bits.

21. (Previously presented): A method of switching between a predetermined plurality of instruction sets used by a data processing apparatus, the method comprising:

in response to a first instruction:

(i) accessing a sequence of bits, the sequence of bits having an address portion and an instruction set indicator portion, the instruction set indicator portion having at least one bit that is not part of the address portion of the sequence of bits;

(ii) identifying an instruction set selected from the predetermined plurality of instruction sets based on the instruction set indicator portion of the sequence of bits;

(iii) setting one or more control flags to indicate that a current instruction set for the data processing apparatus is the instruction set identified based on the instruction set indicator portion of the sequence of bits; and

retrieving a second instruction from an address derived from the address portion of the sequence of bits.

22. (Previously presented): The method of claim 21, further comprising executing the second instruction as an instruction of the current instruction set.

23. (Previously presented): The method of claim 21 in which the predetermined plurality of instruction sets comprises a first instruction set and a second instruction set, and wherein

instructions of the first instruction set are X-bit instructions and instructions of the second instruction set are Y-bit instructions, where Y is different from X.

24. (Previously presented): The method of claim 23 wherein X is 32 and Y is 16.

25. (Previously presented): The method of claim 21 wherein the instruction set indicator portion of the sequence of bits comprises one or more least significant bits of the sequence of bits.

26. (Previously presented): The method of claim 21 wherein the instruction set indicator portion of the sequence of bits comprises one or more most significant bits of the sequence of bits.

27. (Previously presented): A data processing apparatus capable of operating using instructions from a predetermined plurality of instruction sets, the data processing apparatus comprising:

(i) a processor core responsive to a first instruction to access a sequence of bits, the sequence of bits having an address portion and an instruction set indicator portion, the processor core deriving an address of a second instruction from the address portion of the sequence of bits and using the instruction set indicator portion of the sequence of bits to set one or more control flags; and

(ii) a controller responsive to the one or more control flags, the state of the one or more control flags specifying a current instruction set selected from the predetermined plurality of instruction sets, to cause the processor core to execute the second instruction as an instruction from the current instruction set,

wherein the one or more control flags are set without regard to the address derived from the address portion of the sequence of bits.

28. (Previously presented): The apparatus of claim 27 wherein the one or more control flags comprise one or more predetermined bits in a program counter.

29. (Previously presented): The apparatus of claim 27, further comprising a memory system, wherein the memory system is not responsive to the one or more control flags.

30. (Previously presented): The apparatus of claim 27, further comprising a memory system wherein the one or more control flags are not provided to the memory system.

31. (Previously presented): The apparatus of claim 27 in which the predetermined plurality of instruction sets comprises a first instruction set and a second instruction set, and wherein instructions of the first instruction set are X-bit instructions and instructions of the second instruction set are Y-bit instructions, where Y is different from X.

32. (Previously presented): The apparatus of claim 31 wherein X is 32 and Y is 16.

33. (Previously presented): A data processing apparatus capable of operating using instructions from a predetermined plurality of instruction sets, the data processing apparatus comprising:

(i) a processor core responsive to a first instruction to access a sequence of bits to derive an address of a second instruction, the sequence of bits having an address portion and an instruction set indicator portion and the instruction set indicator portion having at least one bit that is not part of the address portion of the sequence of bits, the address of the second instruction being derived from the address portion of the sequence of bits;

(ii) the processor core using the instruction set indicator portion of the sequence of bits to set one or more control flags, the state of the one or more control flags specifying a current instruction set selected from the predetermined plurality of instruction sets; and

(iii) a controller responsive to the one or more control flags to cause the processor core to execute the second instruction as an instruction from the current instruction set.

34. (Previously presented): The apparatus of claim 33 wherein the one or more control flags comprise one or more predetermined bits in a program counter.

35. (Previously presented): The apparatus of claim 33, further comprising a memory system, wherein the memory system is not responsive to the one or more control flags.

36. (Previously presented): The apparatus of claim 33, further comprising a memory system wherein the one or more control flags are not provided to the memory system.

37. (Previously presented): The apparatus of claim 33 in which the predetermined plurality of instruction sets comprises a first instruction set and a second instruction set, and wherein instructions of the first instruction set are X-bit instructions and instructions of the second instruction set are Y-bit instructions, where Y is different from X.

38. (Previously presented): The apparatus of claim 37 wherein X is 32 and Y is 16.

39. (Previously presented): A data processing architecture capable of operating using instructions from a predetermined plurality of instruction sets, the data processing architecture comprising:

(i) a processor core responsive to a first instruction to access a sequence of bits, the sequence of bits having an address portion and an instruction set indicator portion, the processor core deriving an address of a second instruction from the address portion of the sequence of bits and using the instruction set indicator portion of the sequence of bits to set one or more control flags; and

(ii) a controller responsive to the one or more control flags, the state of the one or more control flags specifying a current instruction set selected from the predetermined plurality of instruction sets, to cause the processor core to execute the second instruction as an instruction from the current instruction set,

wherein the one or more control flags are set without regard to the address derived from the address portion of the sequence of bits.

40. (Previously presented): The data processing architecture of claim 39 wherein the one or more control flags comprise one or more predetermined bits in a program counter.

41. (Previously presented): The data processing architecture of claim 39, further comprising a memory system, wherein the memory system is not responsive to the one or more control flags.

42. (Previously presented): The data processing architecture of claim 39, further comprising a memory system wherein the one or more control flags are not provided to the memory system.

43. (Previously presented): The data processing architecture of claim 39 in which the predetermined plurality of instruction sets comprises a first instruction set and a second instruction set, and wherein instructions of the first instruction set are X-bit instructions and instructions of the second instruction set are Y-bit instructions, where Y is different from X.

44. (Previously presented): The data processing architecture of claim 43 wherein X is 32 and Y is 16.

45. (Previously presented): A data processing architecture capable of operating using instructions from a predetermined plurality of instruction sets, the data processing architecture comprising:

(i) a processor core responsive to a first instruction to access a sequence of bits to derive an address of a second instruction, the sequence of bits having an address portion and an instruction set indicator portion and the instruction set indicator portion having at least one bit that is not part of the address portion of the sequence of bits, the address of the second instruction being derived from the address portion of the sequence of bits;

(ii) the processor core using the instruction set indicator portion of the sequence of bits to set one or more control flags, the state of the one or more control flags specifying a current instruction set selected from the predetermined plurality of instruction sets; and

(iii) a controller responsive to the one or more control flags to cause the processor core to execute the second instruction as an instruction from the current instruction set.

46. (Previously presented): The data processing architecture of claim 45 wherein the one or more control flags comprise one or more predetermined bits in a program counter.

47. (Previously presented): The data processing architecture of claim 45, further comprising a memory system, wherein the memory system is not responsive to the one or more control flags.

48. (Previously presented): The data processing architecture of claim 45, further comprising a memory system wherein the one or more control flags are not provided to the memory system.

49. (Previously presented): The data processing architecture of claim 45 in which the predetermined plurality of instruction sets comprises a first instruction set and a second instruction set, and wherein instructions of the first instruction set are X-bit instructions and instructions of the second instruction set are Y-bit instructions, where Y is different from X.

50. (Previously presented): The data processing architecture of claim 49 wherein X is 32 and Y is 16.

51. (Previously presented): A data processing apparatus capable of switching between a predetermined plurality of instruction sets, the data processing apparatus comprising:

(i) means for accessing a sequence of bits in response to a first instruction, the sequence of bits having an address portion and an instruction set indicator portion;

(ii) means for identifying an instruction set selected from the predetermined plurality of instruction sets based on the instruction set indicator portion of the sequence of bits in response to the first instruction;

(iii) means for setting one or more control flags to indicate that a current instruction set for the data processing apparatus is the instruction set identified based on the instruction set indicator portion of the sequence of bits in response to the first instruction; and

(iv) means for retrieving a second instruction from an address derived from the address portion of the sequence of bits in response to the first instruction,

wherein the instruction set identified by the instruction set portion of the sequence of bits is identifiable without regard to the address derived from the address portion of the sequence of bits.

52. (Previously presented): The data processing architecture of claim 51 wherein the one or more control flags comprise one or more predetermined bits in a program counter.

53. (Previously presented): The data processing architecture of claim 51, further comprising a memory system, wherein the memory system is not responsive to the one or more control flags.

54. (Previously presented): The data processing architecture of claim 51, further comprising a memory system wherein the one or more control flags are not provided to the memory system.

55. (Previously presented): The data processing architecture of claim 51 in which the predetermined plurality of instruction sets comprises a first instruction set and a second instruction set, and wherein instructions of the first instruction set are X-bit instructions and instructions of the second instruction set are Y-bit instructions, where Y is different from X.

56. (Previously presented): The apparatus of claim 55 wherein X is 32 and Y is 16.

57. (Previously presented): A method of operating a data processing apparatus, the method comprising:

(i) receiving a first instruction from a first instruction set selected from a predetermined plurality of instruction sets;

(ii) translating the first instruction to generate a first set of one or more control signals;

(iii) accessing a sequence of bits comprising an address portion and an instruction set indicator portion in response to the first set of one or more control signals, the instruction set indicator portion having at least one bit that is not part of the address portion of the sequence of bits;

(iv) setting one or more control flags based upon the value of the instruction set indicator portion of the sequence of bits to specify that a current instruction set is a second instruction set selected from a predetermined plurality of instruction sets;

(v) retrieving a second instruction using an address derived from the address portion of the sequence of bits; and

(vi) translating the second instruction as an instruction from the current instruction set to generate a second set of one or more control signals.

58. (Previously presented): The method of claim 57 wherein the predetermined plurality of instruction sets consists of two instruction sets.

59. (Previously presented): The method of claim 57 wherein the first instruction set consists of X-bit instructions and the second instruction set consists of Y-bit instructions, Y being different from X.

60. (Previously presented): The method of claim 58 wherein the first instruction set consists of X-bit instructions and the second instruction set consists of Y-bit instructions, Y being different from X.

61. (Previously presented): The method of claim 59 wherein X is 32 and Y is 16.

62. (Previously presented): The method of claim 60 wherein X is 32 and Y is 16.

63. (Previously presented): The method of claim 59 wherein X is 16 and Y is 32.

64. (Previously presented): The method of claim 60 wherein X is 16 and Y is 32.

65. (Previously presented): A program counter register comprising:

an ordered set of bits;

wherein a subset of the ordered set of bits identifies an address of an instruction; and at least one bit of the ordered set of bits identifies an instruction set; and

wherein the at least one bit is not a member of the subset.

66. (Previously presented): A method of selecting an instruction set comprising the steps of:

receiving a branching instruction written in a first instruction set of a plurality of instruction sets;

pursuant to the branching instruction, inserting the address of a next instruction into a register and setting the value of a flag, where the value of the flag is not dependent upon the address of the next instruction;

selecting an instruction set based upon the value of the flag; and

acquiring the next instruction at the address inserted into the register.

67. (Previously presented): A processing apparatus comprising:

a pointer for identifying an address of a next instruction that is written in a first

instruction set of a plurality of instruction sets; and

a flag for identifying the first instruction set;

wherein:

the pointer and the flag are both written in response to an instruction from a second instruction set of the plurality of instruction sets, and

the value of the flag is not dependent upon the address of the next instruction.

68. (Previously presented): The apparatus of claim 67 wherein:

the first instruction set is different from the second instruction set.

69. (Previously presented): The apparatus of claim 67 wherein:

the pointer and the flag are located in a single register.

70. (Previously presented): The apparatus of claim 67 wherein:

the pointer and the flag are not located in a single register, yet are written to as if portions of a single register.

APPENDIX B

APPENDIX OF EVIDENCE

None.

APPENDIX C

APPENDIX OF RELATED PROCEEDINGS

Attached, please find excerpts from the Court's claim construction ruling in ARM Ltd. v. picoTurbo, Inc., C.A. No. 4:00-cv-00957-CW (N.D. Cal.). Additional materials related to this action were provided to the Examiner in an Information Disclosure Statement filed on May 2, 2002.

IN THE UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

ARM LIMITED,

Plaintiff,

v.

PICOTURBO, INC.,

Defendant.

/

No. 00-00957 CW

ORDER CONSTRUING DISPUTED CLAIMS AND TERMS

Plaintiff Advanced RISC Machines, Ltd. (ARM) has sued Defendant picoTurbo, Inc. (picoTurbo) alleging infringement of patents held by ARM, specifically U.S. Patent Nos.: 5,740,461 ('461 patent); 5,568,646 ('646 patent); 5,758,115 ('115 patent); 6,021,265 ('265 patent); 5,386,563 ('563 patent); 5,583,804 ('804 patent); and 5,701,493 ('493 patent). The parties dispute the proper definition of several terms and phrases that are used in the patents. ARM and picoTurbo each ask the Court to adopt its proposed construction of the disputed terms and phrases. The matter was heard on April 6, 2001. Having considered the parties' papers, the evidence cited therein and oral argument, the Court construes the disputed terms and phrases as set forth herein.

BACKGROUND

Each of the seven ARM patents at issue in this case relates to reduced instruction set computer (RISC) microprocessors. A microprocessor is a type of computer chip that performs arithmetic, logic and control operations with the assistance of internal memory. These computer chips are in turn comprised of substructures that operate together to transform software instructions into specific functions. RISC microprocessors can operate at high speed by relying on a less complex instruction set or sets than would be required by other systems. The appeal of these microprocessors is their ability to minimize size and power consumption while maximizing operating speed.

The patents at issue here address specific hardware designs of microprocessors and their substructures, as well as methods of using those designs. Four of the seven patents, the '461 patent, the '646 patent, the '115 patent and the '265 patent, relate to the hardware structures within the microprocessor that permit instruction sets of different bit widths (i.e. data sizes) to be employed by the same microprocessor. Two patents, the '563 patent and the '493 patent, relate to the structures that handle "exceptions"--abnormal events that occur during normal processing operations. A seventh patent, the '804 patent, describes a "multiplier-accumulator" structure within a microprocessor that executes a certain kind of arithmetic operation.

DISCUSSION

I. Legal Standard

The interpretation of patent claims is a question of law to be

decided by the Court. See Markman v. Westview Instruments, Inc., 517 U.S. 370, 371-73 (1996). To interpret the claims of a patent, the Court must consider the language of the claims, the patent specification and the prosecution history. See Markman v. Westview Instruments, Inc., 52 F.3d 967, 979 (Fed. Cir. 1995) (citations omitted), aff'd, 517 U.S. 370 (1996).

In construing a claim, the Court must look first to the specific words of the claim. See Vitronics Corp. v. Conceptronic, Inc., 90 F.3d 1576, 1582 (Fed. Cir. 1996). Words in the claim are given their ordinary meaning unless the inventor gives special meaning to them in the specification or prosecution history. See id. The specification may act as a dictionary when it defines terms in the claims, whether expressly or impliedly. See id. However, if the inventor gives a term a special meaning, that meaning must be reasonably clear and used consistently within the patent itself. See Lear Siegler, Inc. v. Aeroquip Corp., 733 F.2d 881, 889 (Fed. Cir. 1984).

Interpretation of a disputed claim term also requires reference to the other claims. See Southwall Tech., Inc. v. Cardinal IG Co., 54 F.3d 1570, 1579 (Fed. Cir. 1995). Each claim term must be interpreted consistently in all claims. See id. Ordinarily, the language of one claim should not be interpreted so as to make another claim, such as a claim dependent on the first claim, identical in scope. Claims should be interpreted to render all of the limitations in the claim meaningful. See Unique Concepts, Inc. v. Brown, 939 F.2d 1558, 1562 (Fed. Cir. 1991). Nevertheless, "[w]hether or not claims differ from each other, one

can not interpret a claim to be broader than what is contained in the specification and claims as filed." Tandon Corp. v. United States ITC, 831 F.2d 1017, 1024 (Fed. Cir. 1987).

Claims must be read in light of the specification. See Markman, 52 F.3d at 979. Claims, however, are not limited to the preferred embodiment described in the specification. See SRI Int'l v. Matsushita Elec. Corp. of Am., 775 F.2d 1107, 1121 (Fed. Cir. 1985) (in banc, plurality opinion).

No other evidence is required if the language of the patent and its prosecution history clearly set forth the meaning of a term. See Vitronics, 90 F.3d at 1582-83. However, extrinsic evidence may be received in order to explain scientific principles and terms. See Markman, 52 F.3d at 980. Extrinsic evidence may also be used to demonstrate what would have been known to one skilled in the art at the time of the invention. See id. Finally, extrinsic evidence may assist the court in ascertaining "'the true meaning of the language employed' in the patent." See id. (quoting Seymour v. Osbourne, 78 U.S. 516, 546 (1871)). Extrinsic evidence may not be employed to support an interpretation that contradicts the plain language of the claims. See Markman, 52 F.3d at 981.

If the language of the claims, the specification and the prosecution history do not provide a clear definition for a claim limitation, the notice function of the patent claim is best served by construing the claim narrowly. See Athletic Alternatives, Inc. v. Prince Mfg., Inc., 73 F.3d 1573, 1580-81 (Fed. Cir. 1996).

The Patent Act allows for a claim to be expressed using means-plus-function language, providing that an element in a patent

claim may be expressed as a means or a step for performing a specified function without the recital of structure, material or acts described in the specification and equivalents thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof. See 35 U.S.C. § 112, ¶ 6. The applicant need not recite a structure in the claim's limitation; however, "[t]he applicant must describe in the patent specification some structure which performs the specified function." Valmont Indus., Inc. v. Reinke Mfg. Co., Inc., 983 F.2d 1039, 1042 (Fed. Cir. 1993). Furthermore, the applicant is limited "to the structure, material, or acts in the specification and their equivalents." Id.

Whether sufficient structure is disclosed in the specification to satisfy the requirements of 35 U.S.C. § 112, ¶ 6 is determined based on the understanding of one skilled in the art. See Atmel Corp. v. Information Storage Devices, Inc., 198 F.3d 1374, 1378-80 (Fed. Cir. 1999). A party claiming that an applicant has failed to set forth an adequate disclosure, and therefore that a patent claim is indefinite, must demonstrate indefiniteness by clear and convincing evidence. See Al-Site Corp. v. VSI Int'l, Inc., 174 F.3d 1308, 1323 (Fed. Cir. 1999) (citing 35 U.S.C. § 282) ("Issued patents have a strong presumption of validity in infringement proceedings. Hence, an accused infringer who defends on grounds of patent invalidity bears the burden of showing patent invalidity by clear and convincing evidence.").

II. Claim Construction

As a preliminary matter, the Court notes that throughout the

briefing of the various patents at issue, picoTurbo argues that certain claims are indefinite. Many of the claims picoTurbo asserts are indefinite are presented as means-plus-function elements and are subject to § 112, ¶ 6. In general, the Court will not decide such issues of indefiniteness at this time. Where a structure is indicated in the patent specification, the Court construes for claim construction purposes that structure to be the structure claimed. Because of the lack of expert testimony as to what a person skilled in the art would understand the disclosed structure to be, and thus the patent to be claiming, the Court will not now rule on the issue of indefiniteness where such testimony would be necessary.

~~A. U.S. Patent No. 5,740,461 ('461 patent)~~

The '461 patent addresses how a microprocessor may handle different kinds of instruction sets. Instructions are generally provided to a microprocessor in "sets" of instructions, where each instruction in a set is of a fixed length, or data size. For example, one set may be made up of individual instructions (instruction words) that are each sixteen bits in length, while another set may be made up of individual instructions that are each thirty-two bits in length. The '461 patent teaches how one microprocessor may respond to two instruction sets of different bit lengths.

1. first decoder . . . second decoder

Claim 1 of the '461 patent reads, in part:

> (ii) a first decoder that . . . receives selected X-bit program instructions words of the first permanent instruction set . . . and

picoTurbo argues that when one of ordinary skill in the art speaks of an event occurring "over" a "cycle, the term "cycle" has a clear meaning as a standard "clock cycle" unit of measurement, which is uniform and tied to the overall speed of the microprocessor.

picoTurbo's assertion is supported by the language of the patent specification. The '646 patent specification uses the terms "cycle" and "clock cycle" interchangeably to indicate a uniform time unit. See '646 patent at 4:36-7 ("early in the clock cycle in which the first decoding means operates"); 5:36-39 ("successive processing clock cycles are illustrated in which a fetch operation, a decode operation and finally an execute operation are performed"); 5:39-41 ("If the particular instruction so requires (e.g. a multiply instruction), then one or more additional execute cycles may be added."); 5:51-59 ("decode cycle"); and Fig. 4. Furthermore, although the patent expressly contemplates that other functions might take longer than a single clock cycle, see e.g. id. at 4:2-8 (multiple fetch cycles); 5:39-41 (multiple execute cycles), it only contemplates a single decode cycle. Accordingly, the Court construes clause (vi) of Claim 1 to require the decoding to take place over a single clock cycle.

C. U.S. Patent No. 5,758,115 ('115 patent)

The '115 patent describes a data processor that can use a number of different instruction sets of different bit sizes. To do this, the microprocessor must know the address in memory of the instruction and to which instruction set it belongs. A flag called a "T-bit" identifies the instruction's instruction set.

1. program counter register modifier

Claim 1 of the '115 patent states that a "program counter register modifier . . . modif[ies] the contents of said program counter register in response to a current program instruction word." '115 patent at 8:1-3. The program counter register indicates the address of the next program instruction word in data memory to be executed. picoTurbo asserts that the program counter register modifier must be capable of modifying both the next program instruction word's memory address and the indicator bit (the T bit), both of which are found in the program counter register. The T bit indicates to what set of instructions a given instruction belongs and thus tells the processor which instruction set to select.

picoTurbo's construction is unnecessary. It is clear from the language of the patent claims that the program counter register modifier modifies the contents of the program counter register in response to a current program instruction word. The parties do not dispute this construction, but rather whether or not the Court should further construe what the contents of the program counter register are. Claim 1 clearly sets forth the contents of the program counter register modifier. See e.g. '115 patent at 7:54-66 ("the address of a next program instruction word"); 8:4-6 ("indicator bits"). It is unnecessary to do so here. Accordingly, the Court construes the term "program counter register modifier" to mean a device that modifies the contents of the program counter register in response to a current program instruction word.

2. controller

Claim 1 of the '115 patent discloses a "controller, responsive to one or more predetermined indicator bits of said program counter register, for controlling said processor core to execute program instruction words of a current instruction set." '115 patent at 8:4-14. picoTurbo would construe this term to mean a multiplexer that is responsive to one or more indicator bits that are contained within the PC Register and that selects either the first or second instruction decoders to decode a current instruction word based on the state of that indicator bit. ARM would construe this term to mean a device that controls the processor core.

The controller disclosed in the '115 patent must be "responsive to one or more predetermined indicator bits of said program counter register;" this controller controls "the processor core to execute program instruction words of a current instruction set." Id. These limitations are in the claim language itself. The indicator bits of the program counter register indicate the instruction set to be used by the processor.

There is no need to limit the type of controller involved to a "multiplexer." Although a multiplexer is discussed in the preferred embodiment, this is not a means-plus-function element and need not be limited to the preferred embodiment where the claim language is clear. Accordingly, the Court construes the term "controller" to mean a device, responsive to one or more predetermined indicator bits of the program counter register, that controls the processor core to execute program instruction words of a current instruction set.

3. first instruction decoder . . . and a second instruction decoder

Claim 1 of the '115 patent discloses a "first instruction decoder for decoding program instruction words of a first instruction set; and a second instruction decoder for decoding program instruction words of a second instruction set." Again, the parties do not dispute the meaning of decoder, but rather whether these clauses require two separate decoders. The language of the patent claim and the specification is clear that two decoders are disclosed in Claim 1. The Claim recites a "first decoder" and a "second decoder" as separate elements and as alternative destinations for instructions routed by the controller. See '115 patent at 8:15-23. The Description of the Preferred Embodiment states, "In the present embodiment, two instruction sets are used: a first instruction set . . . decoded by the first instruction decoder 100, and a second instruction set . . . decoded by the second instruction decoder." Id. at 4:42-47. Accordingly, the Court construes this claim to disclose two decoders.

D. U.S. Patent No. 6,021,265 ('265 patent)

The '265 patent is a continuation or "divisional" of the '115 patent. It issued from the same parent application and shares the same specifications. Unlike the '115 patent, the '265 patent's independent claim does not include limitations directed to a first and second instruction decoder.

1. logic operable to modify

The parties have agreed that the term "logic operable to modify" in the '265 patent should be construed consistently with

the term "program counter register modifier" in the '115 patent. Thus, the Court construes this term to mean a device that modifies the contents of the program counter register in response to a current program instruction word.

2. processor core controller

The parties have agreed that the term "processor core controller" in the '265 patent should be construed consistently with the term "controller" in the '115 patent. Accordingly, the Court construes this term to mean a device, responsive to one or more predetermined indicator bits of the program counter register, that controls the processor core to execute program instruction words of a current instruction set.

3. memory access controller

Claim 1 of the '265 patent discloses a "memory access controller operable to access program instruction words stored in said data memory, said access controller not being responsive to said one or more indicator bits of said program counter register." '265 patent at 8:21-24. ARM construes this term to mean a device or group of devices that control access to memory, such as a memory address decoder. picoTurbo does not so much dispute ARM's construction of this term, as it disputes that any such structure or device is discussed anywhere else in the patent. picoTurbo asserts that this claim is thus invalid for indefiniteness. As noted above, in construing the claims of these patents, the Court will not consider whether the claim is indefinite at this time, as long as the claim can be construed to some degree. Such an argument may be included in a motion for summary judgment of

invalidity. Accordingly, the Court adopts ARM's construction of this term as a device or group of devices that control access to memory, such as a memory address decoder, and declines to rule at this time on the issue of indefiniteness.

E. U.S. Patent No. 5,386,563 ('563 patent)

The '563 patent concerns how a microprocessor handles an event called an "exception." An exception is an interruption in the normal (also called "main" or "user") processing mode of the microprocessor. The '563 patent provides a way for the microprocessor to address the exception while saving the processing state it was in before the exception occurred, allowing it to return to the operation it was performing when it suspended operations in the main processing mode to handle the exception.

1. exception processing mode

picoTurbo assets that the phrase "exception processing mode" in Claim 1 of the '563 patent should be construed to mean the mode used for handling interrupts. ARM more broadly construes "exception processing mode" to mean a mode for handling events that cause suspension of program execution. ARM argues that the patent specification lists that an exception processing mode may be entered into not only upon the occurrence of a software or externally applied interrupt, as in picoTurbo's construction, but also when the data processing apparatus is reset and when a memory access is aborted, which both cause suspensions of program execution. See '563 patent at 3:58-65. The Court construes the term "exception processing mode" to mean a mode for handling events that cause suspension of program execution.

the restoring function is accomplished "under software control," or, in other words, by an instruction (or instructions). Such instructions are decoded by an instruction decoder. Accordingly, the Court construes the structure disclosed for performing the restoring function to be an instruction decoder.

CONCLUSION

For the foregoing reasons, the Court CONSTRUES the disputed terms and phrases in the foregoing manner.

IT IS SO ORDERED.

Dated: JUN 15 2001

CLAUDIA WILKEN
United States District Judge

Copies mailed to counsel
as noted on the following page

United States District Court
For the Northern District of California